

(19)



THE PATENT OFFICE  
2nd M. S. O. BUILDING  
234/4, ACHARYA JAGADISH CHANDRA BOSE ROAD  
CALCUTTA-700 020.

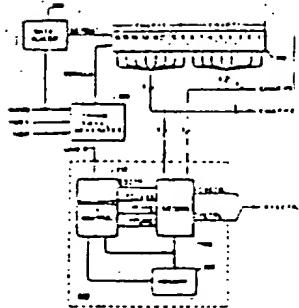
## INDIAN PATENT SPECIFICATION 178674

(51) Int. Cl. : H 04 N 7/087, 7/093	(11) Document No. 178674 Date of Document : 27.4.93.
(52) Ind. Cl. : 186 C	A (42) Date of Publication : 7.6.97.
(21) Application No. : 243/Cal/93.	(71) Applicant : THOMSON CONSUMER ELECTRONICS, INC., of 600 North Sherman Drive, Indianapolis, Indiana 46201, United States of America.
(22) Date of filing : 27.4.93.	(72) Inventor : MICHAEL DAVID LANDIS.
Claims : 02	(74) Agent : H. V. Williams & Co.
Text : 31 Pages; Draw.03. Sheets.	Examiner : A. Srinivas.

(54) Title : AN APPARATUS FOR TELEVISION FOR AUXILIARY VIDEO INFORMATION INCLUDING EXTENDED DATA SERVICES.

## (57) Abstract

An apparatus for television for auxiliary video information comprising extended data services for decoding an extended data services signal in line 21 of field 2. Extended data services provide a general purpose video system information and control capability in addition to basic closed caption operation. Extended data services information is arranged in packets of data. Each packet provides information regarding current or future video programs, the source of the video program, and miscellaneous information such as time of day. The extended data services data may be decoded to control the operation of a video system including a videocassette recorder (VCR) and a television receiver. The apparatus comprising means (200, 220) for receiving television signal and means (230) for processing data words.



PRICE : RUPEES THIRTY ONLY.

**This Page Blank (uspto)**

178674

THE PATENTS ACT. 1970.

COMPLETE

# Specification

SECTION 10

1. The following Specification particularly describes and illustrates the nature of this invention and the manner in which it is to be performed:—

The following Specification particularly describes and illustrates the nature of this invention and the manner in which it is to be performed:—

The present invention relates to an apparatus for television for auxiliary video information including extended data services. A video signal typically includes vertical display intervals, or fields, having a plurality of horizontal line intervals, e.g. 262.5 lines per field in NTSC video systems. The beginning of each vertical and horizontal interval is identified by respective vertical and horizontal sync pulses that are included in a composite video signal. During a portion of each vertical interval, information in the video signal may not be intended for display. For example, a vertical blanking interval spans approximately the first 20 horizontal line intervals in each field. In addition, several line intervals adjacent to the vertical blanking period, e.g. line 21, may be within an overscan region of a video display and will not be visible.

The lack of displayed image information during blanking and overscan intervals makes it possible to insert an auxiliary information component, e.g. teletext or closed caption data, into these intervals. Standards such as Federal Communications Commissions (FCC) Regulations define the format for each type of auxiliary information including the positioning of the information within a vertical interval. For example, the present closed captioning standard (see e.g. 47 CFR 15.119 and 73.682) specifies that digital data corresponding to ASCII characters for closed captioning must be in line 21 of field 1. The FCC specified format provides for two eight-bit digital data words in each occurrence of line 21, field 1.

United States Law requires that all television receivers 13" and larger in size that are sold in the U.S. after 1 July 1993 must be capable of decoding closed caption information (see 47 CFR 15.119). This requirement adds to the cost and complexity of

most televisions. Many television users, particularly individuals who are not hearing impaired, may not wish to utilize the closed caption capability. Thus, television manufacturers must invest in the development of a feature that is of value to only a limited number of individuals who purchase televisions. In addition, many individuals will be compelled to pay for a feature that is of little or no value to them.

Given below is a brief description of the prior art according to Indian Patent Application No. 64/CAL/93 and the distinguishing features of the present application.

Although both inventions relate generally to the same subject matter of processing auxiliary information in a television signal, that is where their similarity ends. The cited prior specification generally concerns a data decoder which operates on bit level of the auxiliary information. The decoder disclosed can determine whether there is any auxiliary data presented in the video signal and whether a particular bit is a zero or a one ( see, e.g. Abstract). Contrary to the present invention, nowhere in the cited specification teaches or suggests any higher level processing or formatting of the information formed by these data bits. There is also no mention of Extended Data Services (EDS) processing in this prior specification.

The claimed invention, on the other hand, relates to the high level processing of auxiliary information such as EDS information at the " data words" level. An aspect of the invention is the problem that is addressed and solved, i.e. the problem of communicating complex auxiliary information via a television signal using a portion of the television signal (e.g. line 21) and a data rate ( e.g. 500 KHz) that impose significant bandwidth limitations. The auxiliary data relates to information that is to be displayed on a display device, e.g. program title. The invention solves that problem for example, by providing processing means that determines control information (e.g. start of data packet) and a class of displayable information ( e.g. current or full--- program) from a single data word as recited in claim 1. The processing means also determines a subclass of information (e.g. program title) from a second data word, and determines information pertaining to the subclass-( eg the actual title information) from a third data word. The claimed invention advantageously provides

178674

efficient use of the available bandwidth permitting communication of complex auxiliary data as required.

*The invention substantially resides in*  
Claim 1, therefore, recites at least the following novel combination of features which are not taught or suggested in the cited prior specification.

According to this invention there is provided an apparatus for television for auxiliary video information comprising extended data services comprising:

means for receiving a television signal including auxiliary information comprising datawords and

mean for processing a first one of said data words for determining control information for controlling processing of said data words and also for determining a class of information to be displayed on a display device; said processing means processing a second one of said data words for determining a subclass of said class of information to be displayed; and said processing means processing a third one of said data words to obtain said information pertaining to said subclass of said class of information to be displayed.

In accordance with the principles of the present invention, a system for processing auxiliary video signals provides for decoding an extended data services signal in line 21 of field 2. Extended data services provide a general purpose video system information and control capability in addition to basic closed caption operation. Extended data services information is arranged in packets of data. Each packet provides information regarding current or future video programs, the source of the video program, and miscellaneous information such as time of day. The extended data services data may be decoded to control the operation of a video system including a videocassette recorder (VCR) and a television receiver.

The invention may be better understood by referring to the drawing, in which:

Figure 1 shows an example of an auxiliary video data signal such as a closed caption or extended data services signal;

Figure 2 shows, in block diagram form, a portion of a video signal processing system incorporating the principles of the invention; and

Figure 3 shows a flowchart illustrating the operation of the system shown in Figure 2.

A decoder for extracting closed caption data is designed to process a closed caption (CC) signal, such as that shown in Figure 1, that occurs during line 21 of each field 1 interval in the video signal. An extended data services (EDS) signal exhibits the same format as that shown in Figure 1, but occurs during line 21 of each field 2 interval. As shown in Figure 1, each line 21 interval in which CC or EDS data is present includes information representing two 8-bit binary words. A system suitable for decoding both CC and EDS information is shown in Figure 2.

In Figure 2, composite video signal VIDEO is input to data slicer 200. Data slicer 200 converts closed caption and extended data services information in analog signal VIDEO into serial digital data in signal SERDAT. Data slicer 200 may be

implemented, for example, using a comparator that compares the level of signal VIDEO to a threshold level. This threshold level is referred to as the slicing level. Logic 0 and logic 1 levels in signal SERDAT represent levels of signal VIDEO that are less than and exceed, respectively, the slicing level.

The CC or EDS data in signal SERDAT is clocked serially into shift register 210 by signal SERCLK. Signal SERCLK is generated by timing signal generator 220 during the data interval within line 21, i.e. the latter portion of line 21 in which the information representing the 16 data bits occurs (see Figure 1). Generator 220 determines when line 21 is present in the video signal by counting horizontal lines in the video signal as indicated by horizontal sync pulses in signal HOR. The horizontal line count is initialized at the beginning of a video field as indicated by a vertical sync pulse on vertical sync signal VERT. Signals VERT and HOR are produced by deflection circuitry in a video system. A phase shift may exist between sync signals produced by the deflection circuitry and the timing of the actual video signal VIDEO. A sync separator in generator 220 produces a separated sync signal from composite video signal VIDEO that is used internally <sup>by</sup> generator 220 to synchronize the generation of signal SERCLK to the actual timing of the data interval within line 21.

The 16 bits of data in register 210 are designated bits 1 to 16 in Figure 3. Bits 8-1 represent the first CC or EDS character, CHAR#1, and bits 16-9 represent the second character, CHAR#2. The serial data in register 210 is converted to parallel data via 16 parallel outputs from register 210. The parallel data is output to other functions in the video system such as an on-screen-display (OSD) processor (not shown in Figure 2) for storing and displaying closed caption data. The parallel data in CHAR#1 and CHAR#2 is also coupled to processing unit 230 for decoding of EDS information.

The format of EDS information is explained in detail below. Briefly, EDS information is arranged in packets of information. Each packet includes a plurality of 8-bit characters from a plurality of occurrences of line 21 of field 2. Each packet represents a complete piece of information that includes both control and data characters. The control characters identify a

particular EDS control function (e.g. start packet, continue packet, or end packet) in a manner that distinguishes EDS information from closed caption information.

EDS control characters also indicate the class and type of information included in a packet. Packet class designations indicate general classifications of the information included in a packet. For example, a packet class indicates whether the packet contains information pertaining to a future program, the current program, the source of a program (e.g. the broadcast network), or miscellaneous information (e.g. time of day). Each packet class encompasses a plurality of specific types of information. In addition to packet class, EDS control characters also identify the particular type of information in a packet. For example, a packet type of "program title" within the "current program" class indicates that the data characters in the packet represent the title of the current program.

In Figure 2, processor 230 includes decoder 235 for detecting and decoding EDS information. The decoding process in decoder 235 is controlled by control unit 233. When line 21 ends as indicated by signal LINE21 from timing generator 230, new character data is present in register 210 and control unit 233 generates signal EDCTRL to initiate the decoding process in decoder 235. Decoder 235 first tests the character bits to determine if the character codes are EDS character codes. If so, decoder 235 then determines if the characters are EDS control characters.

Control characters are further decoded to establish the EDS control function (i.e. start, continue, end), packet class, and packet type. This decoded information is communicated to control unit 233 from decoder 235 via signals PFUNC, PCLASS, and PTYPE, respectively. Control unit 233 may then cause the characters and decoded control information to be stored in memory 237 until a complete packet is received and decoding can be completed.

Decoder 235 also generates system control signals SYSCTRL for controlling the video system in response to the EDS information. For example, a video cassette recorder (VCR) may be activated to begin recording or set the correct time of day in response to control signal VCRCTRL from decoder 235. Similarly, a

television may be controlled via signal TVCTRL to modify the on-screen display (OSD) processor operation to display closed captioning in response to EDS data indicating the presence of closed caption services.

Processor 230 in Figure 2 may also include capability for detecting and decoding closed caption data. More specifically, because EDS information includes control codes that distinguish EDS information from closed caption data, decoder 235 may include separate logic circuits for detecting and decoding closed caption information. In this situation, control signals SYSCTRL generated by processor 230 would include closed caption control signals coupled to the OSD processor to control the closed caption display.

The operation of an embodiment of auxiliary video information processor 230 that includes both closed caption and EDS decoding capability may be better understood by referring to a flowchart shown in Figure 3. Processing begins at step 300 in Figure 3 when closed caption or EDS processing is enabled. Operation halts at step 310 until a line 21 interval is detected. This indication may be provided, for example, by signal LINE21 in Figure 2. At step 320 in Figure 3, serial data from line 21 is loaded into a data register (e.g. register 210 in Figure 2). The bits in the register that represent CHAR#1 in line 21 are tested at step 330 to determine if the character is EDS data. If not, processing continues at step 335 where closed caption characters are detected and processed. Appropriate closed caption control signals are generated at step 335.

As explained below, EDS characters and closed caption characters occur in pairs such that CHAR #1 and CHAR#2 are always either both closed caption data or both EDS data. Thus, when the test of CHAR#1 at step 330 is complete, operation continues at step 340 where CHAR#1 from line 21 is tested to determine if CHAR#1 is an EDS control character. A control character is decoded at step 350 to determine the control function, i.e. start, continue, or end packet, and the packet class. Signals PFUNC and PCLASS shown in Figure 2 are generated during step 350. As described below, when CHAR#1 is an EDS control character, CHAR#2 is a control character that indicates the packet

type. Packet type information is decoded from character 2 at step 360. Signal PTYPE in Figure 2 is generated at step 360.

If the test at step 340 indicates that CHAR#1 is not an EDS control character, the characters are treated as EDS data characters and processed at step 345. The data characters are stored (e.g. in memory 237 in Figure 2) until a complete packet has been received. The data may then be decoded to generate control signals for the system or stored for later use. For example, if the EDS data represents the title of the current program, the title may be stored in memory for subsequent display when activated by a user.

Following closed caption processing at step 335, or EDS information processing at steps 345 and 360, operation continues at step 370 where the system checks to determine if auxiliary video information (i.e. CC or EDS data) processing remains enabled. If enabled, operation returns to step 310 to await the next occurrence of line 21. If disabled, the procedure in Figure 3 is exited at step 390.

The described system processes auxiliary video information formatted in a predetermined manner to facilitate determining whether the information in line 21 is closed caption or EDS information. An exemplary EDS data formatting specification suitable for use with the embodiment depicted in Figure 2 is described below. ~~Aspects of the format described below are incorporated in a "Draft Standard for Line 21 Data Services" that was distributed by the Electronics Industries Association (EIA) Subcommittee 608 on 12 October 1992.~~

## 1. General EDS Data Format Information

The encoding of information for the extended data services (EDS) follows the same general format as for closed caption data encoding. This scheme consists of pairs of characters transmitted in the same field. The characters can be either a control code pair or a data pair. The first byte of the pair determines whether the pair is a control pair or a data pair. If the first byte is in the range of 01h to 0Fh, the pair is a control pair. These values are not defined for captioning or text transmission.

Upon receiving such a control code pair, the decoder would recognize subsequent data as EDS data. This is the same scheme that is used when decoding a closed caption signal to differentiate between caption and text mode, and between operating channel 1 and operating channel 2 (i.e. C1 and C2) of the caption signal. All characters are transmitted using odd parity. This is consistent with the closed caption conventions, and allows for simpler encoding/decoding hardware and software.

There are four varieties of EDS characters: Control, Type, Data, and Checksum. These characters may be transmitted in the combinations shown in Table 1.

Table 1

1 <sup>st</sup> Byte	2 <sup>nd</sup> Byte
Control	Type
Control	Data
Data	Data
Control	Checksum

As described above, the Control byte is in the range of 00h to 0Fh. The Type and Checksum bytes are in the range of 00h to 1Fh. The data byte is in the range of 10h to 7Fh for ASCII data, or in the range of 40h to 7Fh for Non-ASCII data. A Data byte of 00h is a null byte, and is always ignored.

A packet of EDS data is defined to be a collection of these pairs of bytes which conveys a complete piece of information. Each byte of EDS data is associated with a packet of data. A sub-packet is defined to be a control pair followed by some number, maybe zero, of data pairs. A data field is defined to be some number of bits within a Data byte. Each sub-packet may be transmitted independently and surrounded by other information. Note that a full packet could be transmitted using only control pairs, or by also using data pairs for more throughput when possible.

There are three categories of Control bytes: Start, Continue, and End. The Start code indicates the beginning of a packet. The Continue code indicates the following data is part of the packet which began with the last Start code. The End code

indicates the packet is finished. The Type byte always follows the Start code to indicate the type of data contained in the new packet. The Checksum byte always follows the End code and is used for error detection.

Once a packet has been started, the data for the packet can be sent one byte at a time by using a Continue code with each byte to create a separate sub-packet for each byte. Each subpacket occurs during a single instance of line 21. For higher throughput, both bytes during a particular line 21 interval may contain data. In this case, a sub-packet includes data from a plurality of line 21 intervals. The data in a particular line 21 interval belongs to the sub-packet which began with the last Start or Continue code. The transmission of data pairs can not be interrupted by any other information. If it is necessary to interrupt the transmission of data pairs, the transmission of the packet is reestablished by sending a Continue control pair. The example shown in Table 2 illustrates the described process.

Table 2

1 <sup>st</sup> Byte	2 <sup>nd</sup> Byte
... other ...	... other ...
Start	Type
Data	Data
... other ...	... other ...
Continue	Data
Data	Data
Data	Data
... other ...	... other ...
Continue	Data
... other ...	... other ...
End	Checksum
... other ...	... other ...

The described approach will allow broadcasters the flexibility to simultaneously use any combination of captions or text using either C1 or C2, and EDS. It also permits the efficient transmission of EDS information if it is the only service offered on field two of the video signal.

There are four classes of packets currently defined: Current, Future, Network, and Miscellaneous. The Current packet contains information describing the program currently being transmitted. The Future packet contains information about an

upcoming program to be transmitted. The Network packet contains information about the source of the video signal, e.g. the broadcasting network. The Miscellaneous packet contains a variety of other useful information. Table 3 shows the assignment of these packet classes to their respective control codes.

Table 3

Control Code	Function
01h	Current Start
02h	Current Continue
03h	Current End
04h	Future Start
05h	Future Continue
06h	Future End
07h	Network Start
08h	Network Continue
09h	Network End
0Ah	Misc. Start
0Bh	Misc. Continue
0Ch	Misc. End
0Dh	Reserved
0Eh	Reserved
0Fh	Reserved

The transmission of one class of packet may be interrupted by another class of packet because each of the four packet classes has its own group of control codes. As a result, higher priority information can interrupt lower priority information. For example, information about the current program is probably more time critical than is information regarding a future program. A complete packet of "current" information might be sent in the middle of transmitting a packet of "future" information. Thus, single fields of lower priority information can be inserted when unused line 21 intervals are available. However, a packet can only be interrupted by a packet of a different class. This ensures that packets can be "nested" without confusion regarding which packet data is to be associated with when a "continue" control code is issued.

Each packet conveys one piece of information. The first byte of the control code pair that begins a packet (i.e. "start" control code) determines the packet class as shown in Table 3. The type of information contained in the packet is determined by the Type code in byte two of the Start control code pair. The data

bytes associated with a packet are held in temporary storage until the entire packet has been received and the checksum at the end of the packet has been validated. This prevents stored data from being corrupted, and also permits a packet to be aborted in the middle by starting a new packet of the same class.

The data types included in Current and Future packet classes are identical, i.e. the Type designations for both packets is the same. The difference between the Current and Future classes is the "object" of the data, i.e. "current" or "future" program. Any information regarding the current program which can be transmitted via EDS can also be sent in regard to a future program, and vice versa. The data contained in the Future packet always pertains to the "future" program that was most-recently specified in the EDS information. The future program is specified by sending a program identifier as the Type code in a Future packet. This Type code indicates which future program all transmitted information is to pertain to until another program identifier Type code is sent. Similarly, the information in the Current packet class always pertains to the program currently being transmitted. When a new program identifier is sent in a Current packet, the old program has finished and the newly-specified program is beginning.

The data which constitutes the program identifier is simply the scheduled broadcast time, date, and receiver channel number. This has the advantage of being a compact, simple to calculate, and unique identifier for each program broadcast on a given channel per year. Even if the broadcast of a program is delayed, it should still carry its originally scheduled time as its program identifier data throughout its entire broadcast. This will allow the recording of programs which are delayed or run longer than expected. All time and date specifications, including current time and date, are always given as Greenwich Mean Time (GMT). Providing both the start time of a future program and the current program identifier as GMT ensures that the identification of a desired program will be independent of the viewer's time zone and "daylight saving" time status. This permits correct recording, e.g. in a video cassette recorder (VCR), even if the viewer does not tell his VCR what time zone he is in. The only purpose for

specifying the viewer's time zone and daylight saving status is to display the correct local time given the broadcast time as GMT.

## 2. Current and Future Packet Classes

Table 4 shows the assignment of Type codes in the Current and Future packet classes.

Table 4

Type Code	Function
0Ch	Unidentify Program
01h	Program Identifier
02h	Erase Program
03h	Stop Time
04h	Program Title
05h	Program Audience
06h	Program Type
07h	Audio Services
08h	Caption Services
09h	- undefined -
0Ah	- undefined -
78h	Description 8
7Fh	Description 1

### 2.1 "Unidentify Program" Packet Type

This packet contains zero bytes, but indicates the program is to be unidentified. It has the opposite effect of the program identifier packet. When received, all subsequent packets of this class will be ignored until another program identifier is received. This could be used as a signal that the specified program information has all been sent.

### 2.2 "Program Identifier" Packet Type

This packet contains either four or six bytes which define a program start time and date, relative to Greenwich Mean Time, and receiver channel number. The format of each byte is

shown in Table 5. Note that bit #6 in each byte is always set to logic 1 because the information in each byte is not ASCII data.

Table 5

Data	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
minute	1	m <sub>5</sub>	m <sub>4</sub>	m <sub>3</sub>	m <sub>2</sub>	m <sub>1</sub>	m <sub>0</sub>	
hour	1	T	h <sub>4</sub>	h <sub>3</sub>	h <sub>2</sub>	h <sub>1</sub>	h <sub>0</sub>	
day	1	D	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>	
month	1	Z	L	m <sub>3</sub>	m <sub>2</sub>	m <sub>1</sub>	m <sub>0</sub>	
channel	1	c <sub>3</sub>	c <sub>4</sub>	c <sub>5</sub>	c <sub>2</sub>	c <sub>1</sub>	c <sub>0</sub>	
channel	1	s <sub>1</sub>	s <sub>2</sub>	-	-	c <sub>7</sub>	c <sub>6</sub>	

The minute data field has a valid range from 0 to 59, the hour field from 0 to 23, the day field from 1 to 31, and the month field from 1 to 12. The "D" bit determines if daylight saving time is currently being observed throughout the country. The "L" bit determines if the current year is a leap year. The "Z" bit determines if the current time in seconds should be reset to zero. The "D", "L", and "Z" bits are ignored by the decoder when processing this packet (see the description of the "time of day" Type code assignment in the section below that pertains to the Miscellaneous packet class). The "T" bit is processed as part of the program identifier packet to determine if the program is subject to a local tape delay. Even if the broadcast of a program is delayed, it should still carry its originally scheduled time as its program identifier data throughout its entire broadcast.

The "channel" data field is an optional two-byte field having a valid range from 0 to 255. If the channel field is omitted, the receiver channel will default to the currently tuned channel. The channel field allows one channel to specify information for another channel. The channel data field contains a two bit subfield which specifies the source input. The source subfield has a valid range from 1 (S<sub>1</sub>S<sub>0</sub> = 00) to 4 (S<sub>1</sub>S<sub>0</sub> = 11) that can be used in a multi-wire cable system to specify the cable line.

### 2.3 "Erase Program" Packet Type

This packet contains zero bytes, but indicates the specified program data is to be completely deleted. This will be most useful for the Future packet class.

### 2.4 "Stop Time" Packet Type

This packet contains either zero or four bytes which define a program stop time and date relative to Greenwich Mean Time. If the packet contains zero bytes, the existing stop time will be erased. The format of the bytes is the same as for the "program identifier" packet described above in section 2.2 except that no channel data is needed. The "D", "L", and "Z" bits are also ignored by the decoder when processing this packet as described in section 2.2.

### 2.5 "Program Title" Packet Type

This packet contains a variable number, 0 to 32, of bytes which define a program title. If the packet contains zero bytes, the existing program title will be erased. Each byte is an ASCII character in the range of 20h to 7Fh. The variable size of this packet allows for efficient transmission of titles of any length. No "size" indicator byte is needed because the End control code pair is used to terminate the packet.

### 2.6 "Program Audience" Packet Type

This packet contains a variable number of bytes, namely zero to three, which define the intended audience for the program. If the packet contains zero bytes, the existing program audience will be erased. For any data bytes in this packet, bit #6 is set to logic 1 because the data is not ASCII data. The format of the data bytes is shown in Table 6.

178074  
Table 6

b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>4</sub>	b <sub>5</sub>	b <sub>6</sub>	b <sub>7</sub>	b <sub>8</sub>
1	M	W	S	A	T	C	
1	D	--	V	L	N	A	
1	q <sub>2</sub>	q <sub>1</sub>	q <sub>8</sub>	r <sub>2</sub>	r <sub>1</sub>	r <sub>3</sub>	

The data bytes in this packet must be sent in the order shown in Table 6. Table 7 defines the function of the bits for bytes one and two shown in Table 6.

Table 7

Byte One		Byte Two	
C	Children	A	Adult Situations
T	Teens	N	Nudity
A	Adults	L	Language
S	Seniors	V	Violence
W	Women	-	undefined
M	Men	D	Data Services

Bit definitions may be selected from the list in Table 7 in any combination that is needed to communicate the desired information. Byte one indicates the target audience. For example, to specify a program as suitable for the entire family, all bits in byte one would be set. Byte two indicates why the target audience may have been restricted from the entire family.

Byte three in Table 6 contains data fields representing program quality and rating information for movies. The format for byte three is shown in Table 8.

Table 8

q <sub>2</sub>	q <sub>1</sub>	q <sub>0</sub>	Quality	r <sub>2</sub>	r <sub>1</sub>	r <sub>0</sub>	Rating
0	0	0	Unknown	0	0	0	Unknown
0	0	1	-	0	0	1	G
0	1	0	1/2	0	1	0	PG
0	1	1	••	0	1	1	PG-13
1	0	0	•• 1/2	1	0	0	R
1	0	1	•••	1	0	1	NC-17
1	1	0	••• 1/2	1	1	0	X
1	1	1	••••	1	1	1	Ncra

## 2.7 "Program Type" Packet Type

This packet contains a variable number, 0 to  $N$ , of bytes which specifies the type of information included in a particular program. The information in this packet could be used by a viewer to selectively look for certain types of programs. If the packet contains zero bytes, the existing program type will be erased. The first two bytes are not ASCII data and, therefore, bit #6 is set to logic 1 in the first two bytes. The third through  $N$ th bytes are ASCII characters in the range of 20h to 7Fh.

The format of the first two bytes is shown in Table 9.

Table 9

b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
1	N	S	E	E	L
1	f <sub>1</sub>	f <sub>0</sub>	t <sub>1</sub>	t <sub>0</sub>	s <sub>1</sub>

The first two bytes must be sent in the order shown in Table 9.

The first byte defines the general category of the information in the program. The type of information indicated by the bits in the first byte is shown in Table 10.

Table 10

C	Classified
L	Life / Style
E	Education
E	Entertainment
S	Sports
N	News

A logic 1 in a bit position shown in Table 9 indicates that the program provides the corresponding type of information listed in Table 10. If necessary, multiple bits can be set to logic 1 to indicate that the program includes multiple categories of information. Byte two provides additional program information as shown in Table 11.

178074

Table 11

t <sub>1</sub> , t <sub>2</sub>	Format	t <sub>1</sub> , t <sub>2</sub>	Time Slot	s <sub>1</sub> , s <sub>2</sub>	Status
0 0	Special	0 0	Once	0 0	Premiere
0 1	Series	0 1	Once Week	0 1	Live
1 0	Mini-Series	1 0	Weekdays	1 0	Tape Delay
1 1	Movie	1 1	Every Day	1 1	Re-Run

Bytes three to N provide additional information which can be used to further specify the type of programming. These bytes are sent as ASCII characters, but the character codes represent the words listed in Table 12.

Table 12

Code	Qualifier	Code	Qualifier	Code	Qualifier
20h	Unknown	40h	Gymnastics	60h	Singing
21h	Action / Adventure	41h	Health	61h	Skating
22h	Amatuer	42h	Hobby	62h	Skating
23h	Animated	43h	Hockey	63h	Scan Opera
24h	Arts	44h	Home	64h	Soccer
25h	Auto	45h	Horror	65h	Special
26h	Awards	46h	Hunting / Fishing	66h	Talk
27h	Baseball	47h	Improvement	67h	Tennis
28h	Basketball	48h	Informational	68h	Text
29h	Bowling	49h	Instructional	69h	Track / Field
2Ah	Boxing	4Ah	Investment	6Ah	Travel
2Bh	Business	4Bh	Legal	6Bh	Variety
2Ch	Cartoon	4Ch	Local	6Ch	Video
2Dh	Classical	4Dh	Medical	6Dh	Volleyball
2Eh	Comedy	4Eh	Military	6Eh	War
2Fh	Commercial	4Fh	Musical	6Fh	Western
30h	Computer	50h	Mystery / Suspense	70h	World
31h	Concert	51h	National	72h	Wrestling
32h	Contemporary	52h	Nature	73h	special 1
33h	Criminal	53h	Opinion	74h	special 2
34h	Dancing	54h	Performing	75h	special 3
35h	Diving	55h	Personal	76h	special 4
36h	Documentary	56h	Political	77h	special 5
37h	Drama	57h	Professional	78h	special 6
38h	Erotica	58h	Public	79h	special 7
39h	Fantasy	59h	Racing	7Ah	special 8
3Ah	Fiction	5Ah	Religious	7Bh	special 9
3Bh	Financial	5Bh	Romance	7Ch	special 10
3Ch	Football	5Ch	Sales	7Dh	special 11
3Dh	Fund Raiser	5Dh	Science	7Eh	special 12
3Eh	Game / Quiz	5Eh	Service	7Fh	special 13
3Fh	Golf	5Fh	Shopping		

The bytes listed in Table 12 may be used in whatever combination is necessary to specify the desired level of information regarding the type of programming. However, multiple bytes should be sent in proper grammatical order. In addition, it should be noted that receivers may impose limitations on the number of bytes that will be recognized.

The byte designated "Unknown" in Table 12 is the default value if no other bytes from Table 12 are included in the Program Type packet. The twelve "special" bytes listed in Table 12 may be defined by each network to best suit individual programming needs (see the Type code assignments for "special qualifiers" in the section below that describes the Network packet class). The byte designated "Other" in Table 12 indicates that the type of programming is known and does not fit into any of the defined programming types. All twelve of the "special" bytes listed in Table 12 implicitly include the "Other" designation.

### 2.8 "Audio Services" Packet Type

This packet contains either zero or two bytes which define the contents of the main and second audio programs that are associated with the video signal. If the packet contains zero bytes, existing audio services information will be erased. Bit #6 is set to logic 1 in the data bytes in this packet because the data bytes are not ASCII data. The format of the bytes is shown in Table 13.

Table 13

Data	b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>4</sub>	b <sub>5</sub>	b <sub>6</sub>	b <sub>7</sub>
main	1	1 <sub>2</sub>	1 <sub>3</sub>	1 <sub>4</sub>	1 <sub>5</sub>	1 <sub>6</sub>	1 <sub>7</sub>
sap	1	1 <sub>2</sub>	1 <sub>3</sub>	1 <sub>4</sub>	1 <sub>5</sub>	1 <sub>6</sub>	1 <sub>7</sub>

Each of the two bytes listed in Table 13 contains two data fields: language and type. The language field of each byte represents the languages listed in Table 14.

178678

Table 14

$t_2\ t_1\ t_0$	Language
0 0 0	Unknown
0 0 1	English
0 1 0	Spanish
0 1 1	French
1 0 0	German
1 0 1	Japanese
1 1 0	Other
1 1 1	None

The type field of each byte listed in Table 13 is encoded to represent the information shown in Table 15.

Table 15

Main Audio Program	
$t_2\ t_1\ t_0$	Type
0 0 0	Unknown
0 0 1	Mono
0 1 0	Simulated Stereo
0 1 1	True Stereo
1 0 0	Stereo Surround
1 0 1	Data Service
1 1 0	Other
1 1 1	None

Second Audio Program	
$t_2\ t_1\ t_0$	Type
0 0 0	Unknown
0 0 1	Mono
0 1 0	Descriptive Video Service
0 1 1	Non-program Audio
1 0 0	Special Effects
1 0 1	Data Service
1 1 0	Other
1 1 1	None

## 2.9 "Caption Services" Packet Type

This packet contains a variable number, 0 to 8, of bytes which define the available forms of caption encoded data. If the packet contains zero bytes, existing information regarding caption serves will be erased. One byte is included to specify each available service. Bit #6 is set to logic 1 in each byte because the data is not ASCII data. Each of the bytes is in the format shown in Table 16.

Table 16

$b_8$	$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$
1	$l_2$	$l_1$	$l_0$	F	C	T		

The language data field (L2-L0 in Table 16) is encoded using the same format as for the audio services packet described in section 2.8 above. The "F" bit determines if the data is in TV field one ("F" = 0), or in field two ("F" = 1). The "C" bit determines if the data is in channel C1 ("C" = 0), or in channel C2 ("C" = 1). The "T" bit determines if the data is captioning ("T" = 0), or text ("T"=1). This information permits the broadcaster to completely specify the line 21 services that are available.

#### 2.10. "undefined" Packet Types

Type Codes 09h and 0Ah in Table 4 are undefined. These type codes may be defined in the future to further expand EDS capability. For example, one of the undefined type codes might be allocated to provide information regarding video "scrambling". Various approaches are used for encoding, or scrambling, a video signal to prevent viewing by unauthorized users, e.g. "pay-per-view" programming. Information regarding the type of scrambling might be useful to permit authorized users to more effectively decode the scrambled signal.

Another possible use for the undefined codes is to provide information regarding the aspect ratio of the video image in a program. Aspect ratio information would permit the system to select only certain aspect ratio programs. Alternatively, the video receiver could use the aspect ratio information to adapt the signal to the particular display screen aspect ratio of the video receiver.

#### 2.11. "Description 'N'" Packet Type

These packets each contain a variable number, 0 to 32, of bytes which, when combined together, form a description of the program. If the packet contains zero bytes, the existing line of

description information will be erased. Each byte is an ASCII character in the range of 20h to 7Fh. Each packet of this type provides one line of a multiple line description of the program. The description can contain any information the service provider chooses including: episode title, date of release, cast of characters, brief story synopsis, etc. By varying the number of packets of Description "N" type, efficient transmission of program descriptions of any length is possible.

### 3. Network Packet Class

Table 17 lists the assignment of Type codes for the Network packet class.

Table 17

Type Code	Function
00h	Unidentify Network
01h	Network Identifier
02h	Erase All Programs
03h	Network Name
04h	Call Letters
05h	Native Channel
06h	Tape Delay
07h	Special Qualifier 1
12h	Special Qualifier 12

#### 3.1. "Unidentify Network" Packet Type

This packet contains zero bytes and indicates that the network is to be "unidentified". The effect is opposite to that of the "network identifier" packet (see section 3.2 below). After this packet is received, all subsequent packets of the Network class will be ignored until a network identifier packet is received. This packet type can be used as a signal that all network information has been sent.

### 3.2. "Network Identifier" Packet Type

This packet contains either zero or two bytes which define a receiver channel number for which network information is to be specified. The format of the bytes is the same as for the channel data field shown in Table 5 in section 2.2 above. The two byte channel field is optional. The receiver channel will default to the currently tuned channel if not specified. This field allows one channel to specify information for another channel.

### 3.3. "Erase All Programs" Packet Type

This packet contains zero bytes, but indicates all of the program information for the specified network is to be completely deleted.

### 3.4. "Network Name" Packet Type

This packet contains a variable number, 0 to 32, of bytes which define the name of the broadcasting network. If the packet contains zero bytes, the existing network name is erased. Each byte is an ASCII character in the range of 20h to 7Fh. Each network should use a single unique name so that receivers can access information regarding the network that is stored internal to the receiver, e.g. a network logo that can be displayed when a network is selected.

### 3.5. "Call Letters" Packet Type

This packet contains a variable number, 0 to 32, of bytes which define the "call" letters of the local broadcasting station. If the packet contains zero bytes, the existing call letters are erased. Each byte is an ASCII character in the range of 20h to 7Fh.

### 3.6. "Native Channel" Packet Type

This packet contains either zero or two bytes which define the "native" channel number, i.e. local "over-the-air" broadcast channel number, that is assigned to a station. This information is useful if a cable channel number assigned to a station differs from the station's over-the-air broadcast channel number. If the packet contains zero bytes, the existing native channel number is erased. The format of the bytes is the same as for the channel field listed in Table 5 in section 2.2 above.

### 3.7. "Tape Delay" Packet Type

This packet contains either zero or one byte which defines the number of half hours the local station routinely tape delays network programs. If the packet contains zero bytes, the existing tape delay information is erased. The data is not ASCII data so bit #6 is always set to logic 1. The format of the data byte in this packet is shown in Table 18.

Table 18

b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
1	S	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>

The delay field (d<sub>4</sub>-d<sub>0</sub> in Table 18) has a valid range from 0 to 31, which represents time values from 0 hours and 0 minutes to 15 hours and 30 minutes in 30 minute increments. The "S" bit is a sign bit, and determines if the delay value is to be added to the expected program start time ("S" = 0) or subtracted from it ("S" = 1). This delay would apply to all programs on the channel which have the "T" bit set in their program identifier information (see Table 5 in section 2.2 above). The delay value defaults to zero if not specified.

### 3.8. "Special Qualifier 'N'" Packet Type

These packets each contain a variable number, 0 to 32, of bytes which define the text to be associated with the "special" bytes listed in Table 12 that may be used for specifying program information. If the packet contains zero bytes, the text associated with a "special" byte is erased. Each byte in this packet type is an ASCII character in the range of 20h to 7Fh. Each packet provides text for one network specific "special" program information byte. For example, a station which offers mostly sports may define its first five "special" bytes to represent sports such as: Poker, SCUBA, Hang Gliding, Americas Cup, and Olympics. However, a station which offers mostly music may define its first five "special" bytes to represent types of music such as: Heavy Metal, Rap, Pop, Country, and Disco. The meaning of the "special" bytes can be redefined by the network at any time. If no text is received to define a "special" byte, the byte will default to a single blank space.

### 4. Miscellaneous Packet Class

Table 19 lists the assignment of Type codes for the Miscellaneous packet class.

Table 19

Type Code	Function
01h	Time of Day
02h	Time Zone
03h	Line Number
04h	No EDS
05h	Single EDS
06h	Directory EDS
07h	Program Pause
08h	Program Resume
09h	Impulse Capture

#### 4.1. "Time of Day" Packet Type

This packet contains four data bytes which define the current time of day and date relative to Greenwich Mean Time.

The format of the bytes is the same as that shown in Table 5 for the "program identifier" packet (see section 2.2 above), except that no channel data is needed. The "D" bit is used to determine if daylight savings time is currently being observed throughout the country. This information, along with the viewer's specified time zone and whether daylight savings time is locally observed, is used to determine the correct local time. Local time is only used to display the local time for the viewer. All internal timers and clocks should be kept in Greenwich Mean Time.

The "L" bit is used to determine if the current year is a leap year. This is needed to determine if the local day is February 28th or 29th when it is March 1st Greenwich Mean Time. The "Z" bit is used to determine if the current time in seconds should be reset to zero. This allows the time of day to be correct without transmitting the full six bits of data to specify the current number of seconds. The "T" bit is used to determine if the program is subject to a local tape delay. If this bit is set, the time of day clock should not be updated.

#### 4.2. "Time Zone" Packet Type

This packet contains one byte which defines the viewer's time zone and daylight savings status. The data is not ASCII data so bit #6 is always set. The format of the single data byte is shown in Table 20.

Table 20

b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
1	D	h <sub>4</sub>	h <sub>3</sub>	h <sub>2</sub>	h <sub>1</sub>	h <sub>0</sub>	

The hour data field (bits h<sub>4</sub>-h<sub>0</sub> in Table 20) has a valid range from 0 to 23 and represents the nominal delay in hours relative to GMT. The "D" bit determines if daylight savings time is to be observed. This packet should only be sent when all possible viewers reside in the same time zone.

#### 4.3. "Line Number" Packet Type

178674

This packet contains one byte which defines the current line number and field for the tuned channel. This data is not ASCII data so bit #6 is always set. The format of the byte is shown in Table 21.

Table 21

b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
1	F	1	1	1	1	1	1

The "line" field (bits L4-L0 in Table 21) has a valid range from 7 to 31. The "F" bit determines if the data is in TV field one ("F" = 0) or in field two ("F" = 1).

#### 4.4. "No EDS" Packet Type

This packet contains zero bytes and indicates that the tuned channel has no extended data services information available.

#### 4.5. "Single EDS" Packet Type

This packet contains zero bytes and indicates that the tuned channel has extended data services information available for a single channel.

#### 4.6. "Directory EDS" Packet Type

This packet contains zero bytes and indicates that the tuned channel has extended data services information available for multiple channels. This information would be used to identify a station which provides a continuous directory of information about other channels.

#### 4.7. "Program Pause" Packet Type 178074

This packet contains zero bytes and indicates that the current program on the tuned channel has been interrupted. It will need to be retransmitted at least once per minute to maintain a pause. This is because receivers will time-out after one minute even if no program resume packet is sent.

#### 4.8. "Program Resume" Packet Type

This packet contains zero bytes and indicates that the current program on the tuned channel has resumed. It is used to immediately end a program pause. Receivers should perform an automatic program resume if one has not been received within the last minute following a program pause.

#### 4.9. "Impulse Capture" Packet Type

This packet contains either zero, eight, or ten bytes which define a program stop time and date, and start time and date, all relative to Greenwich Mean Time, and receiver channel number. If the packet contains zero bytes, existing information regarding impulse capture is erased. The format of the bytes is the same as for the "stop time" (see section 2.4) followed by the "program identifier" (see section 2.2). This packet provides all the information needed to permit a program to be easily recorded. The program identifier bytes follow the stop time bytes because the program identifier contains a variable number of bytes. The "D", "L", and "Z" bits are ignored by the decoder when processing this packet. The receiver channel will default to the currently tuned channel if not specified.

Various modifications of the features described above are possible. For example, future modifications of FCC requirements may permit EDS data to be included in video lines other than line 21 of field 2. In addition, processing unit 230 in Figure 2 may be implemented using a microprocessor. In this case, the operation illustrated in Figure 3 may be implemented in

178074

software. Alternatively, a combination of hardware and software may be utilized. Also, the embodiment shown in Figure 2 may be implemented in one integrated circuit that includes the data slicer, data register, and decoding functions. These and other modifications are intended to be within the scope of the following claims.

---

30

WE CLAIM :

1. An apparatus for television for auxiliary video information comprising extended data services comprising;

means ( 200, 220 ) for receiving a television signal ( VIDEO ) including auxiliary information, <sup>said</sup> ~~such~~ apparatus characterised by; said auxiliary information <sup>said</sup> ~~such~~ comprising data words; and

means ( 230 ) for processing a first one of said data words for determining control information for controlling processing of said data words and also for determining a class of information to be displayed on a display device; said processing means ( 230 ) processing a second one of said data words for determining a subclass of said class of information to be displayed; and said processing means ( 230 ) processing a third one of said data words to obtain said information pertaining to said subclass of information to be displayed.

2. An apparatus for television for auxiliary video information comprising extended data services substantially as herein described and illustrated.

DATED THIS 27th DAY OF APRIL, 1993.

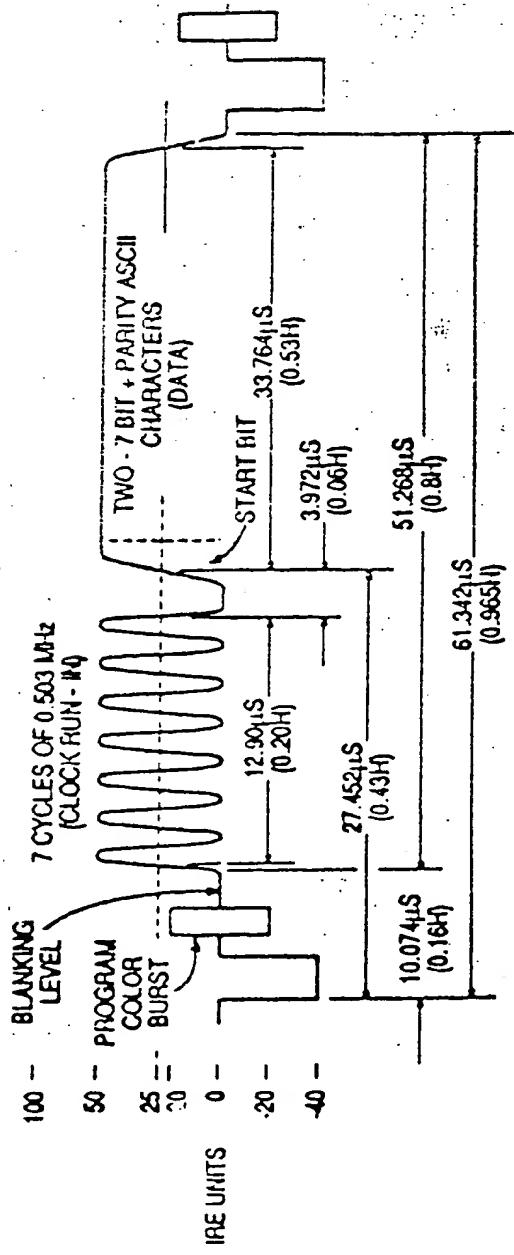
L. S. DAVAR  
( L.S.DAVAR )  
OF L. S. DAVAR & CO.,  
APPLICANT'S AGENT.

THOMSON CONSUMER ELECTRONICS, INC.

3 SHEET  
SHEET

178674

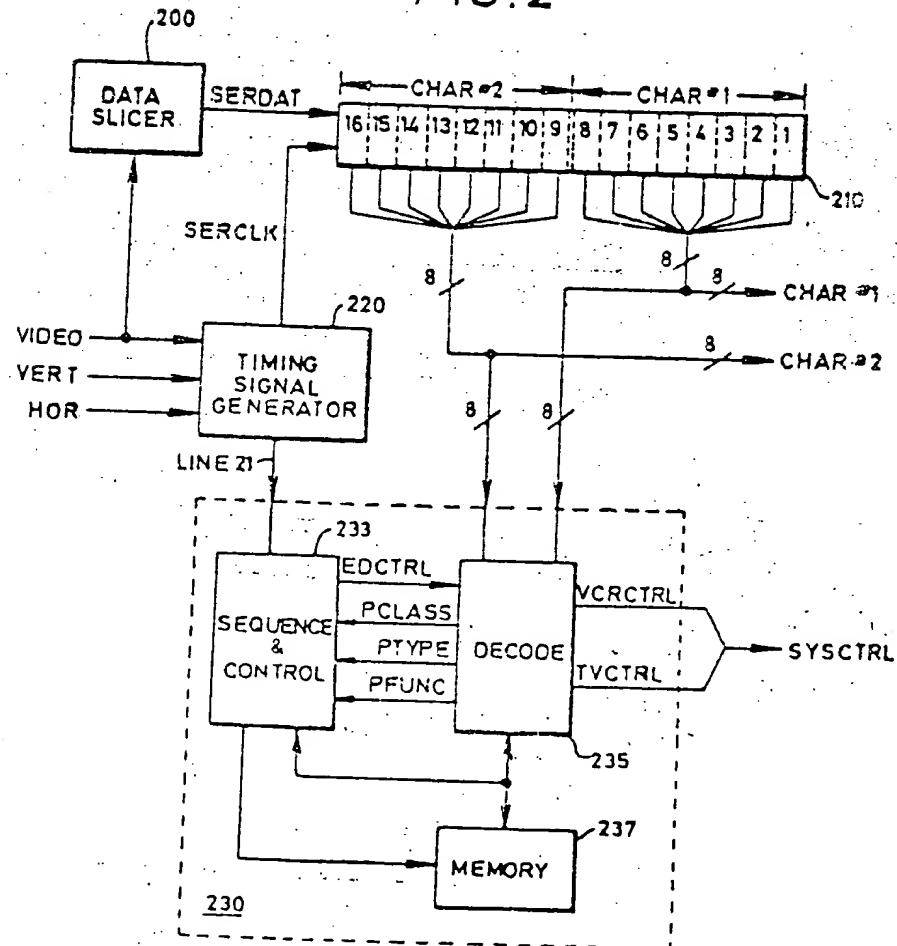
FIG. 1



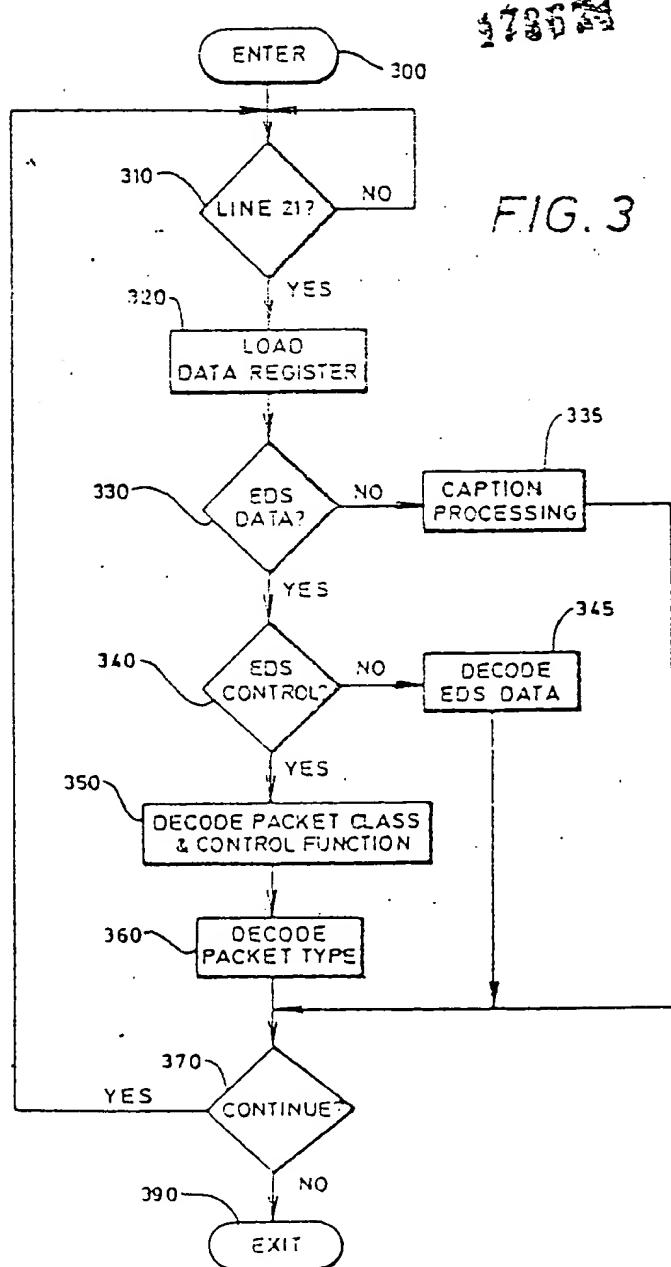
U.S. DAVAR  
(L.S. DAVAR)

OF M.V. WILLIAMS & CO.  
MANUFACTURERS' AGENT

FIG. 2



*Chen*  
(C. S. D. Chen)  
OF M. U. WILLIAMS & CO.  
APPLICANTS' AGENT



✓ William  
(C.S. Danner)  
O.F. H. U. WILLIAMS & CO  
MANUFACTURERS - IMPORTERS

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

**This Page Blank (uspto)**